Validity Check of Giannakis' Formula

Advanced Digital Signal Processing

Christos Samaras

1 Non-Gaussian character of input signal

First, signal x[k] is constructed using MATLAB as an output of a MA-q process with coefficients of $[1.0 \quad 0.93 \quad 0.85 \quad 0.72 \quad 0.59 \quad -0.10]$, driven by white non-Gaussian noise v[k]. Noise v[k] follows an exponential distribution with mean value of 1. In order to get better results, mean value of noise is subtracted during output construction. Generally, non-zero skewness indicates non-Gaussian behavior. Skewness is calculated via the following mathematical relationship

$$\gamma_3^{\mathrm{v}} = \frac{\sum_{i=1}^{N} (\mathrm{v}(i) - \hat{m_{\mathrm{v}}})^3}{(N-1)\hat{\sigma}_{\mathrm{v}}^3}$$

and is found equal to 1.899376, confirming the non-Gaussian character of input v[k].

2 3^{rd} order cumulant estimation

3^{rd} order cumulants $c_3^x(\tau_1, \tau_2)$ of x[k] are estimated using "cumest" function from HOSA Toolbox via the indirect method and plotted below.

.

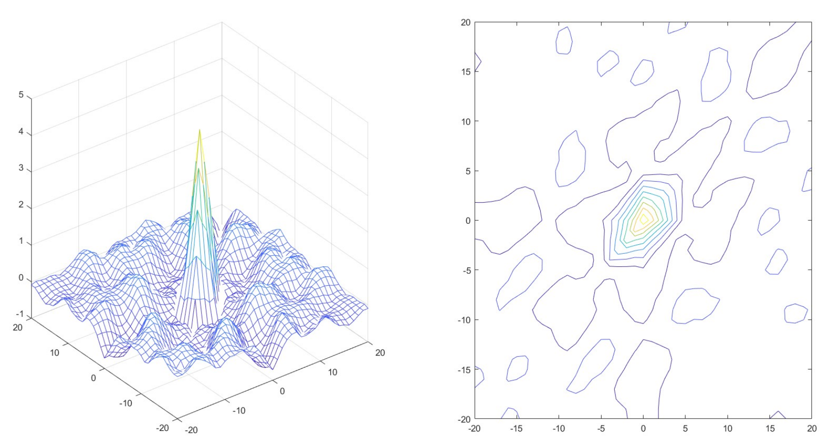


Figure 1: 3D and 2D plots of 3^{rd} order cumulants

3 Estimation of MA system Impulse Response

Using the estimated $c_3^x(\tau_1, \tau_2)$, now the impulse response $\hat{h}[k]$ of the MA system is estimated through Giannakis' formula. Based on this formula,

$$\hat{h}[k] = \frac{c_3^x(q,k)}{c_3^x(q,0)} \quad \text{, k=0,1,2,..., q}$$

$$\hat{h}[k] = 0 \quad \text{, k>q}$$

Thus, considering the (21,21) element of the 41x41 matrix that arises from the $c_3^x(\tau_1, \tau_2)$ estimation as $c_3^x(0,0)$ and MA-q order equal to 5, impulse response h[k] is found as

$$\hat{h}[k] = [1.000 \quad 0.8591 \quad 0.7521 \quad 0.7394 \quad 0.6864 \quad 0.5222]$$

4 Giannakis' formula considering sub/sup estimation

Giannakis' formula is now implemented in combination with sub-estimation ($q = 3$) and sup-estimation ($q = 8$). Therefore, impulse response of former occasion $\hat{h}_{sub}[k]$ is

$$\hat{h}_{sub}[k] = [1.000 \quad 0.8692 \quad 0.7024 \quad 0.6020]$$

and impulse response of latter occasion $\hat{h}_{sup}[k]$ is

$$\hat{h}_{sup}[k] = [1.0000 \quad 0.7779 \quad 0.7682 \quad 0.6824 \quad 0.4840 \quad 0.6996 \quad 0.9646 \quad 1.4158 \quad 1.5318]$$

5 Normalized Root Mean Square Error

Estimated output is calculated via convolution of the input v[k] and the estimated impulse response $\hat{h}[k]$. Then, NRMSE is computed using the following formula

$$NRMSE = \frac{RMSE}{max(x[k]) - min(x[k])} \quad \text{where} \quad RMSE = \sqrt{\frac{\sum_{k=1}^{N}(x_{est}[k] - x[k])^2}{N}}$$

and is found equal to 0.053745. Here are the plots of the constructed output and estimated output.

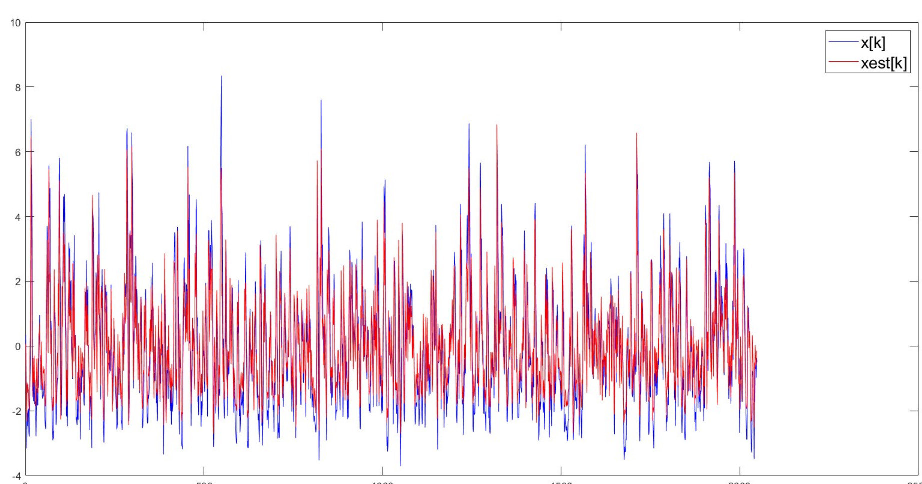


Figure 2: Original x[k] vs estimated $x_{est}[k]$

6 NRMSE for sub/sup estimation

Next, NRMSE is found for sub-estimation and sup-estimation. Here is a table containing the errors for each estimation

Experiment	MA order	NRMSE
Estimation	5	0.053745
Sub-estimation	3	0.084406
Sup-estimation	8	0.216455

Table 1: NRMSE for each estimation

Sub-estimation error is 57.05% above estimation error. As for sup-estimation, error is 302.74% larger than estimation's error and 156.45% larger than NRMSE of sub-estimation. Without NRMSE values being known, a logical thought would have been that the additional terms would help to approximate the original system with greater accuracy. In fact, after error calculations, it was found that extra terms which theoretically introduce extra information, lead to much greater error. In contrast, sub-estimation approaches the initial system with a minor error compared to sup-estimation. After all, Giannakis' original formula provides the best result. Further down, the sub-estimated and sup-estimated output signals are displayed.

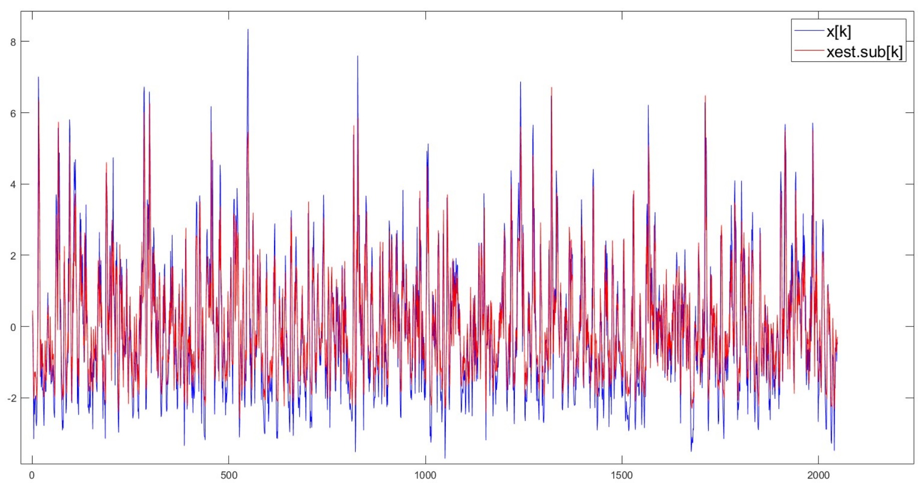


Figure 3: Original x[k] vs sub-estimated $x_{est}[k]$

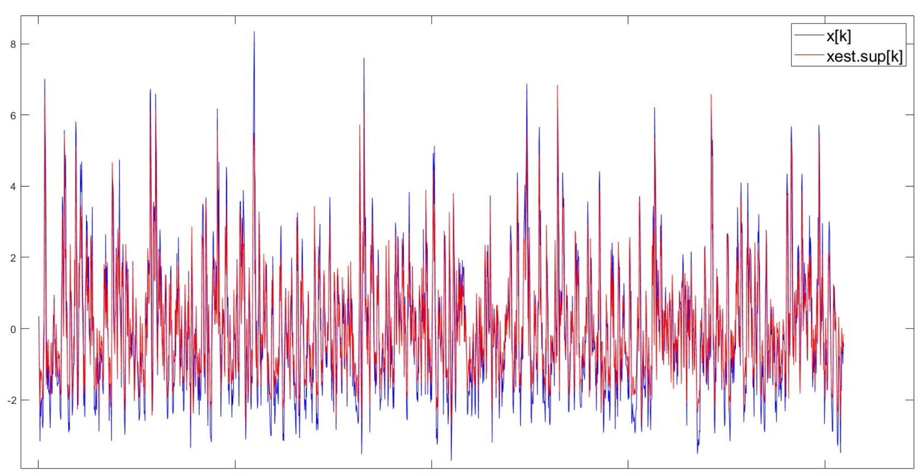


Figure 4: Original x[k] vs sup-estimated $x_{est}[k]$

7 Additive White Gaussian Noise

Analysis is being repeated, but this time AWGN is interfered in output $x[k]$ leading to a new output signal $y[k]$. Noise produces a variation in the Signal-To-Noise-Ratio (SNR) from $30dB$ to $-5dB$ with a step of $-5dB$, so new output is $y_i[k] = x[k] + n_i[k]$. NRMSE is calculated for each noise contaminated output $y_i[k]$ for each level of SNR and the following plot is taken.

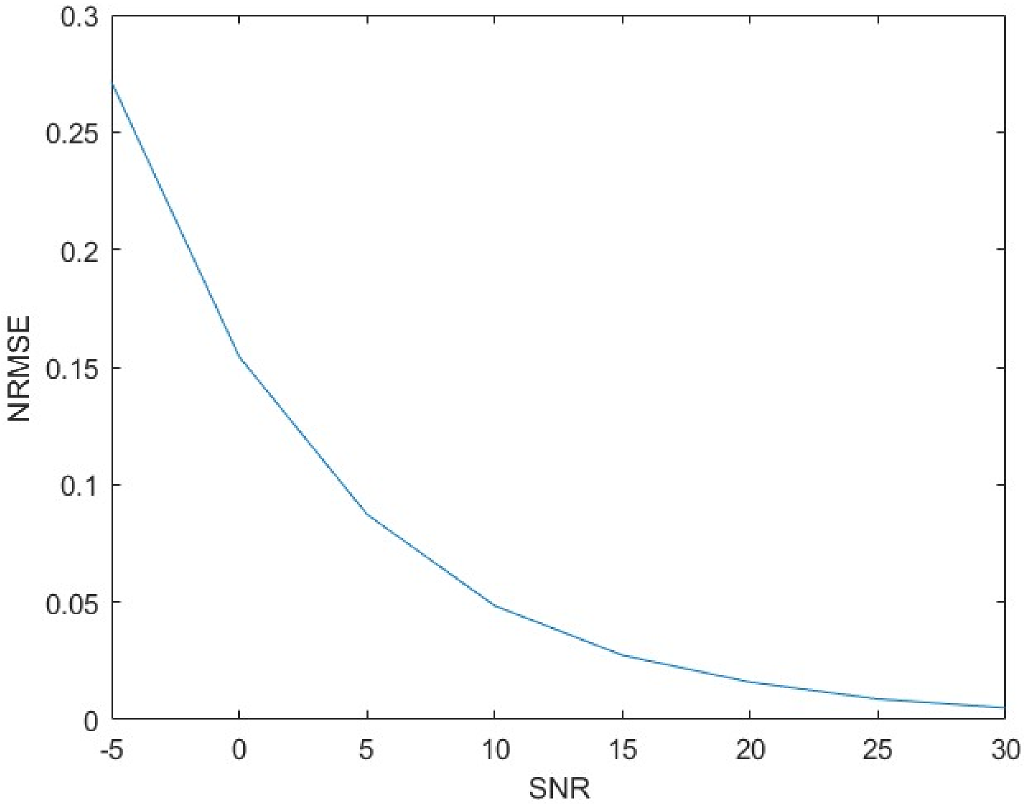
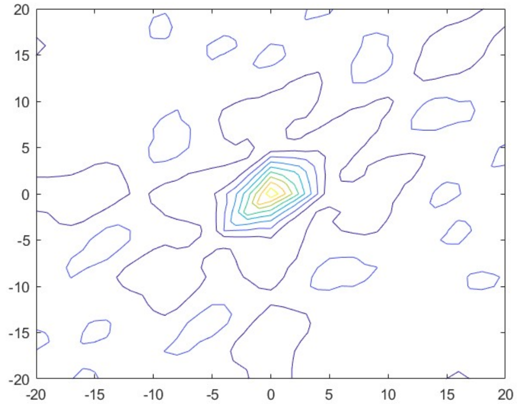


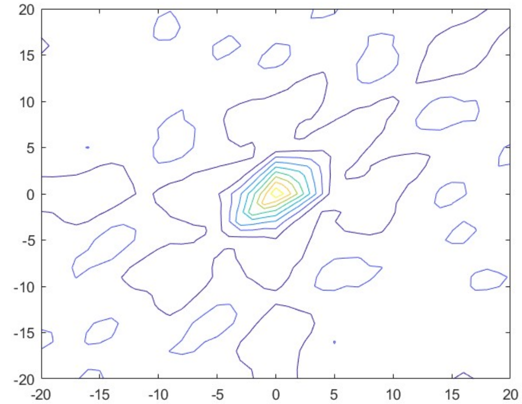
Figure 5: NRMSE with respect to SNR

As seen from graphical representation, normalized root mean square error decreases as the SNR increases.

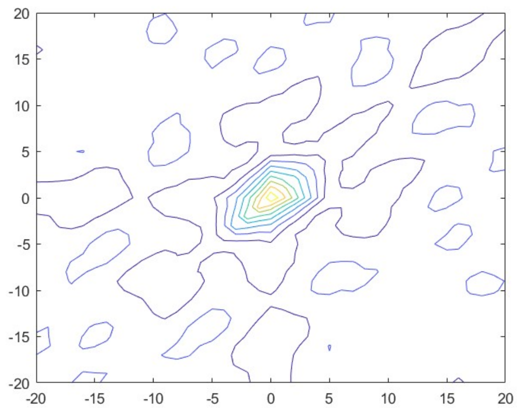
Next, third-order cumulants plots for all SNR values are presented. For high SNR values, i.e. 20dB, 25dB, 30dB, $c_3^y(\tau_1, \tau_2)$ of noise contaminated output does not present significant differences compared to $c_3^x(\tau_1, \tau_2)$ of original output. For SNR values less than $15dB$, cumulants spread is observed in areas where there was no content before. This phenomenon is particularly intense for SNR 0 and -5 dB. At these SNR levels, noise has dramatically contaminate output signal masking its true characteristics and making the 3^{rd} order cumulants plot completely inaccurate and distorted.



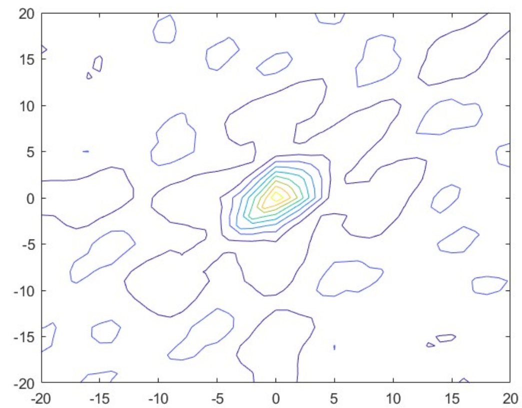
(a) 3^{rd} order cumulants for 30dB SNR



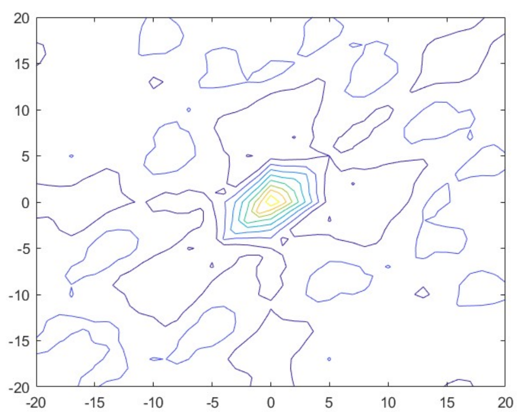
(b) 3^{rd} order cumulants for 25dB SNR



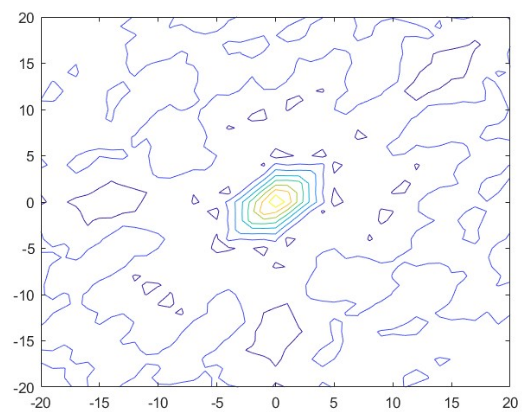
(c) 3^{rd} order cumulants for 20dB SNR



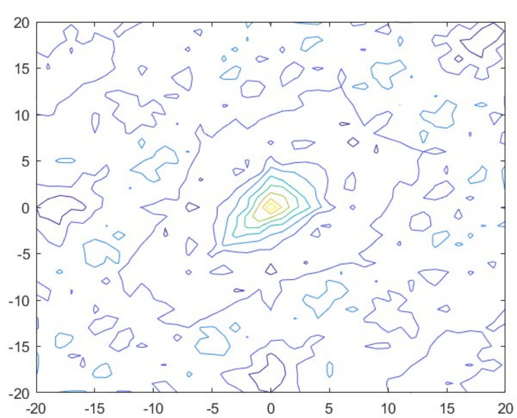
(d) 3^{rd} order cumulants for 15dB SNR



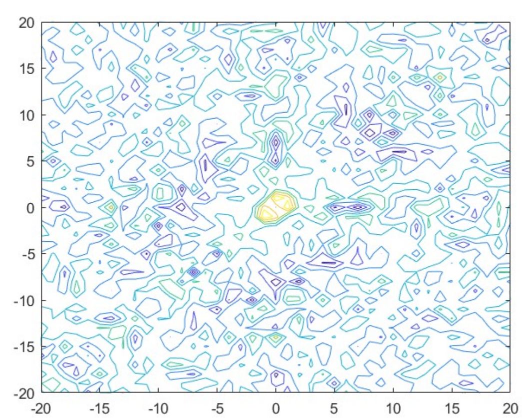
(e) 3^{rd} order cumulants for 10dB SNR



(f) 3^{rd} order cumulants for 5dB SNR



(g) 3^{rd} order cumulants for 0dB SNR



(h) 3^{rd} order cumulants for -5dB SNR

Figure 6: Third-Order Cumulants with SNR variation

8 MATLAB code

Signals were constructed using this code:

```
N=2048;
v_out=exprnd(1, [1, N]);
v_mean=mean(v_out);
v=v_out-v_mean;
save('v_data.mat', 'v');

b = [1.0, 0.93, 0.85, 0.72, 0.59, -0.10];
x=conv(v,b,'same');
save('x_data.mat', 'x');
```

Study was implemented in "exercise3.mlx" file, Here is its code:

```
% ---------------------------------------------------------------------
%                               exercise 3
% ---------------------------------------------------------------------
addpath("C:\Users\samag\OneDrive\ADSP\hosa\hosa")
addpath("C:\Users\samag\OneDrive\ADSP\signal construction")
load('v_data.mat'); load('x_data.mat');

%% ex3.1 - Skewness Calculation
v_mean = mean(v);
v_var = var(v);
skewness = sum((v - v_mean).^3) / (2047 * v_var^(3/2));
fprintf('Skewness= %f\n', skewness);
fprintf('----------------------------------------------------------------')
%% ex3.2 - Third-Order Cumulant
norder = 3; maxlag = 20; nsamp = 64; overlap = 0; flag = 'biased';
cmat = zeros(2*maxlag+1, 2*maxlag+1);
for k1 = -maxlag:maxlag
    cmat(:,k1+maxlag+1) = cumest(x,norder,maxlag,nsamp,overlap,flag,k1);
end

figure(1)
mesh(-maxlag:maxlag, -maxlag:maxlag, cmat)
title('Third-Order Cumulant (3D)')

figure(2)
contour(-maxlag:maxlag, -maxlag:maxlag, cmat, 8)
title('Third-Order Cumulant (2D)')
fprintf('----------------------------------------------------------------')
%% ex3.3 - Estimation using Giannakis formula (q=5)
c3 = cmat(26, 21:26);
h = c3 / c3(1);
fprintf('Estimation')
fprintf('h = '); disp(h);

%% ex3.4 - Sub and Sup Estimation (q=3 and q=8)
c3sub = cmat(24, 21:24);
h_sub = c3sub / c3sub(1);
fprintf('Sub-estimation')
fprintf('h_sub = '); disp(h_sub);

c3sup = cmat(29, 21:29);
h_sup = c3sup / c3sup(1);
fprintf('Sup-estimation')
fprintf('h_sup = '); disp(h_sup);

```

```
fprintf('----------- Comparison of original x with x_est, x_sub, x_sup -----------')
xest = conv(v, h, 'same');
plot_estimation(x, xest, 'X[k]', 'Xest[k]', 3, 'X and Xest using Giannakis formula');
xest_sub = conv(v, h_sub, 'same');
plot_estimation(x, xest_sub, 'X[k]', 'Xest_s_u_b[k]', 4, 'X and Xest using sub-estimated Giannakis formula');
xest_sup = conv(v, h_sup, 'same');
plot_estimation(x, xest_sup, 'X[k]', 'Xest_s_u_p[k]', 5, 'X and Xest using sup-estimated Giannakis formula');

NRMSE = calc_nrmse(x, xest);
fprintf('Estimation NRMSE= %f\n', NRMSE);
fprintf('Sub-estimation NRMSE= %f\n', calc_nrmse(x, xest_sub));
fprintf('Sup-estimation NRMSE= %f\n', calc_nrmse(x, xest_sup));

fprintf('-------------------------------- Task 7 --------------------------------')
SNR = [30, 25, 20, 15, 10, 5, 0, -5];
cmats = zeros(2*maxlag+1, 2*maxlag+1, length(SNR));
NRMSEall = zeros(1, length(SNR));

for i = 1:length(SNR)
    y = awgn(x, SNR(i), 'measured');
    for k1 = -maxlag:maxlag
        cmats(:, k1+maxlag+1, i) = cumest(y, norder, maxlag, nsamp, overlap, flag, k1);
    end

    figure(i+5)
    contour(-maxlag:maxlag, -maxlag:maxlag, cmats(:,:,i), 8)
    title(sprintf('Third-Order Cumulant (SNR=%d dB)', SNR(i)))

    NRMSEall(i) = calc_nrmse(x, awgn(x, SNR(i), 'measured'));
end

figure(14)
plot(SNR, NRMSEall)
title('NRMSE according to SNR')
xlabel('SNR')
ylabel('NRMSE')

%% Functions used:

function NRMSE = calc_nrmse(x, xest)
    RMSE = sqrt(mean((xest - x).^2));
    NRMSE = RMSE / (max(x) - min(x));
end

function plot_estimation(x, xest, legend1, legend2, fig_num, plot_title)
    figure(fig_num)
    plot(x, 'color','blue')
    hold on
    plot(xest, 'color', 'red')
    legend(legend1, legend2, 'Location', 'northeast', 'FontSize', 16);
    hold off
    title(plot_title)
end
```